

BAB II

LANDASAN TEORI

Pada bab II ini semua materi dari dasar-dasar teori yang digunakan dalam penelitian ini akan dibahas oleh penulis. Hal-hal yang mencakup materi tentang : Penelitian terdahulu, penjelasan huruf Hiragana dan Katakana, Media Pembelajaran, API Gesture, dan API Voice Recognition, serta Android semua akan dibahas pada bab ini,

2.1 Kajian Penelitian Terdahulu

Beberapa hasil penelitian yang terkait dengan penelitian ini adalah sebagai berikut:

- a. Adhi, 2013. Aplikasi Belajar Hiragana Jepang berbasis Android Skripsi Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Gunadarma. Hasil dari penelitian ini adalah sebuah aplikasi android Pengenalan huruf Hiragana yang didalamnya berisi tentang materi yang berkaitan dengan huruf Hiragana, pengenalan huruf Hiragana dasar dan lambang *bunyi sei-on, daku-on, dan handaku-on*, beserta beberapa kata dengan menggunakan huruf Hiragana, dan kuis yang berisis 20 soal. Pada penelitian ini tidak menggunakan API Gesture maupun API Voice Recognition, kekurangan pada penelitian ini adalah tidak adanya fitur untuk berlatih menulis huruf Hiragana serta tidak adanya fitur untuk user berlatih pelafalan huruf Hiragana. Pada penelitian ini materi yang disampaikan hanya huruf Hiragana saja.
- b. Hambali, at all, 2013. Rancang Bangun Aplikasi Pembelajaran Aksara Jawa Berbasis Android .Journal sistem Informasi STIMIK STIKOM Surabaya. Hasil dari penelitian ini adalah sebuah aplikasi pembelajaran Aksara Jawa yang memiliki fitur berlatih menulis aksara Jawa dengan mengimplementasikan API Gesture Android. Kekurangan aplikasi ini adalah materi yang dimiliki tentang aksara jawa, berbeda dengan bahasa Jepang yang memiliki beraneka ragam jenis huruf.

- c. Badawi, 2012. Aplikasi Pengenalan Huruf Hiragana dan Katakana Berbasis Android Skripsi Ilmu Komputer, Universitas Gunadarma, hasil dari penelitian ini adalah sebuah aplikasi pengenalan huruf Hiragana dan Katakana berbasis Android, tetapi aplikasi ini hanya dapat dijalankan pada sistem operasi android 2.1 (Éclair) sampai 2.3 (Gingerbread), aplikasi ini hanya memberikan pengetahuan tentang pengenalan dari huruf dasar Hiragana dan Katakana, dimana dari setiap jenisnya jumlah huruf baik dari Hiragana maupun Katakana masing-masing memiliki 46 huruf, disertai dengan contoh pelafalan huruf dan tata cara penulisan huruf. Kekurangan pada aplikasi ini adalah tidak adanya fitur kuis untuk menguji kemampuan pengguna, selain itu aplikasi ini hanya dapat dijalankan pada sistem operasi Android 2.1(Eclair) sampai 2.3(Gingerbread).
- d. Astamar, 2012. Aplikasi Pengenalan Bahasa Jepang Berbasis Android. Skripsi jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Gunadarma. Hasil dari penelitian ini adalah aplikasi Android Pengenalan Bahasa Jepang yang berisi pengenalan tentang huruf kana dasar/asli yang dilengkapi dengan tata cara menulis huruf dengan menggunakan media gambar. Metode penulisan yang digunakan pada penelitian ini adalah *SDLC (System Development Life Cycle)*. Kekurangan pada aplikasi ini adalah aplikasi ini hanya mengenalkan cara membaca dan menulis huruf Hiragana hanya dengan menggunakan gambar, tidak ada fitur untuk berlatih menulis huruf bagi pengguna, serta aplikasi ini hanya dapat dijalankan pada sistem Android versi 2.2 (froyo).

Dari beberapa penelitian tersebut, belum ada penelitian tentang aplikasi media pembelajaran huruf Hiragana dan Katakana dengan mengimplementasikan API Gesture Android dan API Voice Recognition Android, oleh karena hal tersebut maka penelitian ini akan mengembangkan hasil dari penelitian yang pernah dilakukan tersebut dengan membuat aplikasi media pembelajaran huruf Hiragana dan Katakana yang didalamnya terdapat beberapa fitur yaitu :

- 1) Materi tentang huruf Hiragana dan Katakana
- 2) Lambang bunyi huruf Hiragana
- 3) Menulis huruf Hiragana dan Katakana
- 4) Pelafalan huruf Hiragana dan Katakana
- 5) Menulis kata
- 6) Kosakata
- 7) Kuis

2.2 Huruf Hiragana

Huruf Hiragana pertama kali dipakai oleh kaum wanita, huruf wanita (onnmade) atau sekarang lebih dikenal dengan huruf Hiragana memiliki bentuk penulisan yang melengkung, berbeda dengan huruf Katakana yang memiliki bentuk penulisan bersudut tajam. Hal inilah yang membuat huruf Hiragana dianggap lebih sulit untuk ditulis daripada huruf Katakana.

Huruf Hiragana terdiri dari 46 huruf. Huruf-huruf tersebut tersusun berdasarkan 5 huruf vokal yaitu a, i, u, e, o, dan 40 huruf silabel, yaitu kombinasi gabungan dari konsonan dan vokal dengan deretan ka, sa, ta, na, ha, ma, ya, ra, wa, dan 1 huruf konsonan n (Renariah, 2002:10). Asal dari huruf-huruf Hiragana dapat dilihat pada **Gambar 2.1**.



あ (安)	い (以)	う (宇)	え (衣)	お (於)
か (加)	き (幾)	く (久)	け (計)	こ (己)
さ (左)	し (之)	す (寸)	せ (世)	そ (僧)
た (太)	ち (知)	つ (川)	て (天)	と (止)
な (奈)	に (任)	ぬ (奴)	ね (衿)	の (乃)
は (波)	ひ (比)	ふ (不)	へ (部)	ほ (保)
ま (末)	み (美)	む (武)	め (女)	も (毛)
や (也)		ゆ (由)		よ (与)
ら (良)	り (利)	る (留)	れ (礼)	ろ (呂)
わ (和)	ゐ (為)		ゑ (恵)	を (遠)
ん (天)				

Gambar 2.1 Asal Mula huruf Hiragana (Renariah, 2002:10)

2.2.1 Tata Cara Penulisan Huruf Hiragana

Dalam menulis huruf Hiragana (ひらがな), terdapat tata cara urutan penulisan yang benar, sama seperti pada saat menulis huruf Katakana maupun huruf Kanji. Dengan memperhatikan peraturan urutan penulisan maka akan menghasilkan bentuk tulisan yang sesuai, begitu juga sebaliknya apabila tulisan yang dihasilkan berasal dari urutan yang tidak diperhatikan maka akan menghasilkan bentuk yang berbeda, sehingga hal ini dapat membuat huruf yang ditulis menjadi tidak dapat terbaca (Nippon Hoso Kyokai, 2015). Peraturan tata cara urutan penulisan huruf Hiragana dapat dilihat pada **Gambar 2.2**.



Gambar 2.2 Tata cara Urutan Penulisan Huruf Hiragana (Nippon Hoso Kyokai, 2015)

Untuk penulisan garis kesamping atau horizontal, dimulai dari arah kiri ke kanan, untuk garis yang berdiri atau vertikal dimulai dari arah atas ke bawah, sedangkan untuk garis yang melengkung, ditulis dengan memulai sesuai arah jarum jam.

2.2.2 Lambang Bunyi pada Huruf Hiragana

Selain huruf Hiragana dasar, juga terdapat huruf Hiragana yang telah dimodifikasi dengan menambahkan beberapa tanda tertentu atau menggabungkan dengan beberapa huruf tertentu sehingga bunyi yang dihasilkan akan berbeda. Lambang bunyi tersebut yaitu : *sei-on*, *daku-on*, *handaku-on*, dan *yoo-on* (Renariah, 2002).

1) *Sei-on*

Huruf Hiragana yang memiliki jumlah lebih dari 46 disebut dengan lambang bunyi *sei-on* (Renariah, 2002:10-11). Huruf Hiragana yang termasuk dalam lambang bunyi *sei-on* dapat dilihat pada **Gambar 2.3**



あ	い	う	え	お	/a i u e o/
か	き	く	け	こ	/ka ki ku ke ko/
さ	し	す	せ	そ	/sa shi su se so/
た	ち	つ	て	と	/ta chi tsu te to/
な	に	ぬ	ね	の	/na ni nu ne no/
は	ひ	ふ	へ	ほ	/ha hi fu he ho/
ま	み	む	め	も	/ma mi mu me mo/
や	ー	ゆ	ー	よ	/ya - yu - yo/
ら	り	る	れ	ろ	/ra ri ru re ro/
わ	ん	を			/wa o n

Gambar 2.3 Huruf Hiragana lambang bunyi *Sei-on* (Renariah, 2002:10-11)

2) *Daku-on*

Lambang bunyi *daku-on* merupakan lambang bunyi yang berasal dari huruf Hiragana yang mendapat tambahan tanda *tenten* [“]. Lambang bunyi *daku-on* memiliki jumlah 20 huruf. Lambang bunyi ini berasal dari bagian huruf konsonan vokal k, s, t dan h (Renariah, 2002:10-11). Huruf Hiragana yang termasuk dalam lambang bunyi *daku-on* dapat dilihat pada **Gambar 2.4**.

が	ぎ	ぐ	げ	ご	/ ga gi gu ge go /
ざ	じ	ず	ぜ	ぞ	/ za ji zu ze zo /
だ	ぢ	づ	で	ど	/ da ji zu de do /
ば	び	ぶ	べ	ぼ	/ ba bi bu be bo /

Gambar 2.4 Huruf Hiragana lambang bunyi *Daku-on* (Renariah, 2002:10-11)

3) *Handaku-on*

Lambang bunyi *handaku-on* merupakan lambang bunyi yang berasal dari huruf hiragana yang mendapat tambahan tanda maru [°]. Lambang bunyi *handaku-on* memiliki jumlah 5 huruf. Lambang bunyi ini berasal dari bagian huruf konsonan vocal h (Renariah, 2002:10-11). huruf Hiragana yang termaksud dalam lambang bunyi *handaku-on* dapat dilihat pada **Gambar 2.5**.

ぱ ぴ ぷ ぺ ぽ / pa pi pu pe po/

Gambar 2.5 huruf Hiragana lambang bunyi *Handaku-on* (Renariah, 2002:11)

4) *Yoo-on*

Lambang bunyi *yoo-on* merupakan lambang bunyi yang berasal dari huruf Hiragana ki, shi, chi, ni, hi, mi, ri, dan huruf Hiragana lambang bunyi *daku-on* gi, ji, dan pi dengan menambahkan huruf や (ya), ゆ (yu), よ (yo) yang ditulis dengan ukuran lebih kecil daripada ukuran huruf yang sebenarnya. Jika ditulis dengan ukuran huruf yang sebenarnya maka cara membacanya akan berbeda, contoh : ひや (hiya) dan ひゃ (hya). Lambang bunyi *yoo-on* memiliki jumlah 33 huruf (Renariah, 2002:12). Huruf Hiragana yang termaksud dalam lambang bunyi *yoo-on* dapat dilihat pada **Gambar 2.5**.

きゃ	きゅ	きょ	/ kya kyu kyo /
しゃ	しゅ	しょ	/ sha shu sho /
ちゃ	ちゅ	ちょ	/ cha chu cho /
にゃ	にゅ	にょ	/ nya nyu nyo /
ひゃ	ひゅ	ひょ	/ hya hyu hyo /
みゃ	みゅ	みょ	/ mya myu myo /
りゃ	りゅ	りょ	/ rya ryu ryo /
ぎゃ	ぎゅ	ぎょ	/ gya gyu gyo /
じゃ	じゅ	じょ	/ ja ju jo /
びゃ	びゅ	びょ	/ bya byu byo /
ぴゃ	ぴゅ	ぴょ	/ pya pyu pyo /

Gambar 2.6 Huruf Hiragana *yoo-on* (Renariah, 2002:12)

2.2.3 Fungsi Huruf Hiragana

- 1) Sebagai partikel penanda yang berhubungan dengan tata bahasa Jepang. Contoh : カレーを食べた。(Karē wo tabeta) Aksara Hiragana [を] (wo) pada kalimat tersebut berfungsi sebagai penanda objek.
- 2) Berfungsi sebagai konjugasi atau infleksi untuk melengkapi bacaan yang ditulis pada bagian samping huruf kanji, kata kerja, dan kata sifat. Huruf Hiragana pada hal ini juga disebut dengan okurigana (送り仮名) yang memiliki arti “huruf pendamping”. Contoh : 行く (iku), yang berarti “pergi”. Huruf Hiragana く (ku) pada kata tersebut disebut sebagai okurigana.
- 3) Sebagai panduan membaca untuk huruf Kanji yang jarang dipakai, cara membacanya yang tidak beraturan, dan diletakan di atas huruf Kanji, huruf Hiragana pada hal ini juga disebut 振り仮名 (furigana).
- 4) Digunakan untuk menulis kosakata yang huruf Kanjinya tidak umum, atau jarang dipakai, atau huruf Kanjinya susah ditulis. Contoh : dalam bahasa Jepang kata cantik adalah *kirei*, apabila ditulis dengan huruf

Kanji akan menjadi 綺麗 . Huruf tersebut susah untuk ditulis dan tidak umum, sehingga kebanyakan ditulis hanya menggunakan huruf Hiragana menjadi きれい .

- 5) Sebagai huruf pertama yang dipelajari oleh anak-anak ataupun pemula, sebelum mereka mempelajari huruf Kanji.

2.3 Huruf Katakana

Huruf Katakana (カタカナ) merupakan salah satu dari 2 huruf kana pada bahasa Jepang selain huruf Hiragana. Huruf Katakana berjumlah 46 karakter yang melambangkan suku kata. Huruf Katakana berasal dari fragmentasi huruf-huruf Kanji. Mirip dengan huruf Hiragana, huruf Katakana juga merupakan huruf silabis, setiap huruf Katakana akan melambangkan satu bunyi tanpa ada artinya. Huruf Katakana digunakan untuk menuliskan kata-kata bahasa asing seperti bahasa Inggris, bahasa Perancis dan bahasa Jerman, huruf Katakana juga digunakan untuk menuliskan nama-nama penulis dan tempat asing (Renariah, 2002:14). Jenis-jenis huruf Katakana dapat dilihat pada **Gambar 2.7**.

ア (阿)	イ (伊)	ウ (宇)	エ (江)	オ (於)
カ (加)	キ (幾)	ク (久)	ケ (介)	コ (己)
サ (散)	シ (之)	ス (須)	セ (世)	ソ (曾)
タ (多)	チ (千)	ツ (川)	テ (天)	ト (止)
ナ (奈)	ニ (二)	ヌ (奴)	ネ (祢)	ノ (乃)
ハ (八)	ヒ (比)	フ (不)	ヘ (部)	ホ (保)
マ (末)	ミ (三)	ム (牟)	メ (女)	モ (毛)
ヤ (也)		ユ (由)		ヨ (与)
ラ (良)	リ (利)	ル (流)	レ (礼)	ロ (呂)
ワ (和)	ン (尔)			

Gambar 2.7 Asal Mula huruf Katakana (Renariah, 2002:14)

2.3.1 Fungsi Huruf Katakana

- 1) Berfungsi untuk menulis kata serapan atau kata pinjaman yang berasal dari bahasa asing yang disebut dengan 外来語 (*gairaigo*).
Contoh : アイスクリーム (*aisukurīmu*), yang diserap dari bahasa Inggris “Ice Cream” yang berarti “es krim”.
- 2) Berfungsi untuk menuliskan kata-kata atau nama-nama asing, contoh: nama negara, ataupun nama orang, seperti berikut : インドネシア (*Indoneshia*) yang berarti indonesia, ジョン (*john*) untuk nama “John”.
- 3) Berfungsi sebagai penekanan dan penegasan pada sebuah kata.
- 4) Untuk penulisan “onomatope” sebuah kata atau kalimat yang mencontoh bunyi-bunyi yang digambarkan, seperti suara anjing menggonggong ワンワン (“*wanwan*”) dan lainnya.
- 5) Berfungsi untuk penulisan nama produk, merk atau lainnya. Contoh : ホンダ (“*Honda*”).

2.4 Media Pembelajaran

Media pembelajaran adalah sebuah alat maupun metode atau teknik yang bisa digunakan untuk menyajikan dan menyalurkan sebuah pesan dan informasi yang berasal dari sumber belajar kepada penerima pesan belajar, sehingga dapat merangsang pikiran dan perasaan dari peserta didik dan mendorong agar terciptanya sebuah proses interaksi komunikasi edukatif antara guru dan peserta didik secara tepat (Hamalik, 1989 : 12).

Media pembelajaran mempunyai fungsi sebagai alat bantu di dalam suatu proses kegiatan belajar mengajar, peranan media pembelajaran sangat penting, media pembelajaran mempunyai peran sebagai perantara di dalam proses penyampaian pelajaran dari guru kepada peserta belajar. Materi bahan pelajaran yang rumit dapat disampaikan secara sederhana oleh sebuah media belajar.

Media pembelajaran yang baik akan membuat peserta belajar lebih aktif dalam memberikan sebuah tanggapan, maupun umpan balik, dan juga akan

mendorong peserta belajar untuk aktif dalam melakukan kegiatan praktek-praktek belajar yang benar.

2.4.1 Fungsi Media Pembelajaran

Fungsi utama sebuah media pembelajaran adalah sebagai alat bantu dalam sebuah kegiatan pembelajaran. Seiring dengan perkembangan zaman pada saat ini media tidak hanya sebagai alat bantu saja tetapi juga sebagai pembawa informasi sebuah pembelajaran (Hamalik, 1994). Fungsi dari sebuah media pembelajaran adalah sebagai berikut:

- 1) Sebagai alat bantu untuk mewujudkan suasana belajar mengajar yang efektif.
- 2) Sebagai pelengkap proses belajar agar suasana pembelajaran menjadi lebih menarik.
- 3) Untuk mempercepat proses kegiatan belajar mengajar.
- 4) Membantu peserta belajar untuk menangkap atau memahami pengertian yang diberikan oleh guru.
- 5) Untuk melengkapi proses kegiatan belajar mengajar agar menarik minat peserta belajar.
- 6) Sebagai alat untuk mempertinggi mutu kegiatan belajar mengajar.
- 7) Sebagai media pembangkit motivasi anak untuk belajar.

2.4.2 Jenis-Jenis Media

Berdasarkan informasi, media dapat diklasifikasikan dalam beberapa jenis (Arsyad, 2009), yaitu:

- 1) Media Grafis

Merupakan media visual yang menyajikan fakta, ide ataupun gagasan melalui penyajian yang berasal dari kata-kata, kalimat, angka-angka, dan dari simbol maupun gambar. Selain itu grafis juga digunakan untuk menarik perhatian, memperjelas sebuah gagasan ide, dan mengilustrasikan fakta-fakta agar menjadi menarik dan diingat orang.

2) Media Bahan Cetak

Merupakan media visual yang pembuatanya melalui sebuah proses percetakan/printing. Penyampaian pesan akan disampaikan melalui huruf dan gambar-gambar yang diilustrasikan agar pesan atau informasi yang disajikan menjadi lebih jelas.

3) Media Gambar Diam

Merupakan media visual yang berupa gambar, dan berasal dari proses fotografi yang dilakukan.

4) Media Proyeksi Diam

Merupakan sebuah media visual yang memproyeksikan pesan, dimana hasil dari proyeksi tersebut tidak bergerak dan hanya memiliki sedikit unsur gerakan.

5) Media Audio

Merupakan sebuah media penyampaian pesan, dan pesan tersebut hanya dapat diterima oleh indera pendengaran saja. Pesan yang telah disampaikan akan dituangkan ke dalam lambang-lambang auditif yang berupa kata-kata, music, dan sound effect.

6) Media Audio Visual Diam

Merupakan sebuah media penyampaian pesan, dimana pesan tersebut hanya dapat diterima oleh indera pendengar dan indera pengelihatan saja, gambar yang dihasilkan adalah gambar diam dan hanya memiliki sedikit unsur gerak.

2.5 Android

Android merupakan sebuah sistem operasi yang ditujukan untuk sebuah perangkat *mobile* tidak hanya sistem operasi, tetapi Android juga mencakup *middldeware* dan aplikasi. Android dikembangkan oleh Google berdasarkan Linux berdasarkan dengan antarmuka pemrograman Java (Casasola, 2012 : 17).

Android merupakan sistem operasi yang bersifat *open source*. Dengan lisensi di bawah *open source* Apache, yang berarti siapapun yang ingin mengembangkan

ataupun menggunakan Android dapat mendownload sumber kode Android secara penuh.

Kelebihan Android adalah, Android memberikan penawaran pendekatan terpadu pada pengembangan sebuah aplikasi. Pengembang hanya perlu mengembangkan aplikasi untuk Android, selain itu aplikasi yang dibuat oleh pengembang akan dapat dijalankan pada perangkat yang berbeda selama perangkat tersebut menggunakan sistem operasi Android.

2.5.1 Versi Android

Semenjak pertama kali sistem operasi Android telah mengalami beberapa kali pembaharuan, mulai dari perbaikan bug, perubahan desain *user interface* dan penambahan fitur-fitur baru. Inovasi dan penelitian yang dilakukan terus menerus merupakan salah satu dari kunci kesuksesan yang diraih oleh Android. Ciri khas Android untuk menarik hati para pengguna *smartphone* dan *gadget* adalah dengan memberi nama versi-versi sistem operasi miliknya dengan sesuai urutan abjad dan nama makanan, hal ini bertujuan agar para pengguna Android dapat mengingat versi-versi sistem operasi milik Android. Berikut merupakan versi-versi dari sistem operasi milik Android (Triadi, 2013):

1) Android Versi 1.1

Sistem Android yang dikenal dengan nama Android *Alpha* dan Android *Beta*. Diperkenalkan pertama kali pada tahun 2007 dan diaplikasikan pada *smartphone* pertama kali pada tahun 2009 pada pertengahan bulan Maret, dan dikenal sebagai cikal bakal OS Android.

2) Android Versi 1.5 (Cupcake)

Android Cupcake merupakan versi terbaru dari Android, Android Cupcake dirilis pada bulan Mei tahun 2009. Android Cupcake memberikan beraneka ragam kelebihan dibandingkan dengan versi pendahulunya, beberapa fitur terbaru yang dimiliki oleh Android Cupcake yaitu, fitur unggah *video* ke Youtube, *headset bluetooth* yang *nirkabel*.

3) Android Versi 1.6 (Donut)

Android Donut dirilis pada bulan September tahun 2009. Android Donut memiliki beberapa kelebihan dibanding dengan pendahulunya yaitu, memiliki

tampilan untuk indikator baterai, *fitur zoom in zoom out*, dan penggunaan koneksi CDMA, *gesture* dan mesin *text to speech* dan mendukung layar dengan resolusi WVGA.

4) Android Versi 2.0/2.1 (Eclair)

Android Eclair dirilis pada bulan Oktober pada tahun 2009. Perubahan yang dibawa oleh Android Eclair adalah, *pengoptimalan hardware*, peningkatan google map versi 3.1.2, browser baru dengan dukungan HTML5, dan dukungan *Flash* dan *zoom* pada kamera, dan *Bluetooth* dengan versi 2.1.

5) Android Versi 2.2 (Froyo: Frozen Yoghurt)

Android Froyo dirilis pada bulan Mei pada tahun 2010. Fitur-fitur yang dimiliki oleh Android Froyo adalah sebagai berikut:

- Meningkatkan fungsionalitas USB tethering dan hotspot Wi-Fi
- Dukungan pemasangan aplikasi pada penyimpanan eksternal
- Dukungan Adobe Flash
- Dukungan tampilan PPI (hingga 320 ppi), misalnya layar 4" 720p
- Gestur pembesaran pada Galeri

6) Android Versi 2.3 (Ginger Bread)

Android Ginger Bread dirilis pada bulan desember pada tahun 2010, beberapa fitur-fitur baru pada Android Ginger Bread adalah sebagai berikut :

- Dukungan multi kamera pada perangkat, termasuk kamera depan.
- Dukungan bagi pemutar video WebM/VP8, dan audio AAC
- Dukungan bagi resolusi dan ukuran layar ekstra-besar (WXGA dan yang lebih tinggi)

7) Android Versi 3.0/3.2 (Honeycomb)

Android Honeycomb dirilis pada bulan Mei pada tahun 2011, Android Honeycomb hanya khusus bagi para pengguna tablet PC berbasis Android. *User Interface* yang dimiliki juga sangat berbeda dengan *User Interface* milik *smartphone* Android.

8) Android Versi 4.0 (Ice Cream Sandwich)

Android Ice Cream Sandwich dirilis pada bulan Oktober pada tahun 2011. Beberapa fitur baru telah dimiliki oleh Android Ice Cream Sandwich, fitur-fitur baru tersebut adalah sebagai berikut:

- Fitur pendeteksi wajah
- Kualitas video lebih baik
- Penambahan aplikasi pengedit foto bawaan
- WI-FI DIRECT

9) Android Versi 4.1/4.3 (Jelly Bean)

Android Jelly Bean dirilis pada bulan Juli pada tahun 2012, Android jelly Bean memiliki beberapa fitur baru, fitur baru tersebut adalah sebagai berikut:

- Audio multi-saluran
- Peningkatan aplikasi kamera
- Perluasan notifikasi
- Antarmuka pengguna yang lebih halus.

10) Android Versi 4.4 (Kitkat)

Android Kitkat dirilis pada bulan Oktober pada tahun 2013, fitur-fitur baru yang dimiliki oleh Android Kitkat adalah sebagai berikut:

- Peningkatan aksesibilitas API
- Sensor batching, Step Detector, dan Counter API
- Inframerah
- Perekam aktifitas layar yang terintegrasi

11) Android Versi 5.0 (Lollipop)

Android Lollipop dirilis pada bulan Juni pada tahun 2014, fitur-fitur baru yang dimiliki oleh Android Lollipop adalah sebagai berikut:

- Desain antarmuka yang dinamakan “Material Design”.
- 64-bit ART compiler.
- Project volta yang berguna untuk meningkatkan daya hidup baterai 30% lebih lama.

12) Android Versi 6.0 (Marshmallow)

Android Marshmallow dirilis pada bulan Mei pada tahun 2015, fitur-fitur baru yang dimiliki oleh Android Marshmallow adalah sebagai berikut:

- Fitur *fingerprint* atau pengenalan sidik jari untuk membuka kunci pada *smartphone*.
- Mendukung USB Type-C.

13) Android Versi 7.0 (Nougat)

Android Nougat dirilis pada bulan maret pada tahun 2016, fitur-fitur baru yang dimiliki oleh Android Nougat adalah sebagai berikut:

- Unicode 9.0 emoji dan skin modifier support.
- Kemampuan untuk layar zoom.
- Platform virtual reality dunia maya (antarmuka VR).
- Peningkatan fungsionalitas Doze, yang bertujuan untuk memperpanjang masa pakai baterai.

2.5.2 API Gesture

Gesture merupakan sebuah goresan atau coretan yang berasal dari tarikan tangan untuk melakukan suatu hal. Gesture membuat pengguna dapat berinteraksi dengan sebuah aplikasi untuk memanipulasi objek layar. Beberapa contoh *gesture* yang telah diimplementasikan pada Sistem Operasi Android adalah sebagai berikut : *touch, long press, swipe, drag, double touch, pinch open*, dan *pinch close*. Dimulai dari Android versi 1.6, pada SDK platform Android memperkenalkan sebuah aplikasi baru dalam bentuk sebuah emulator bernama Gesture Builder. Gesture Builder memiliki beberapa fungsi, diantaranya adalah : untuk menyimpan, menampilkan, menggambar dan mengenali sebuah gerakan (Casasola, 2012 : 13).

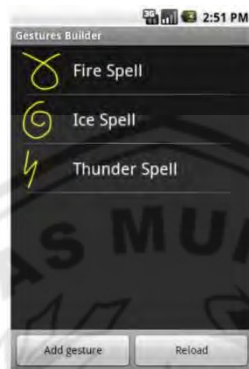
Sistem pengenalan yang dimiliki oleh *gesture* terbagi menjadi beberapa komponen, komponen-komponen tersebut adalah sebagai berikut (Dutta *et al.*, 2012:684):

- 1) “*Template Library*” berfungsi untuk menyimpan sample hasil dari tulisan tangan sebagai objek sikap.
- 2) “*Gesture Builder*” berfungsi untuk membuat sebuah library menggunakan inputan sebuah *gesture* manual, atau menggunakan *sample database* yang telah disediakan sebelumnya.
- 3) “*Recognition*” adalah modul yang berfungsi untuk mengambil tulisan atau goresan tangan yang telah dibuat sebelumnya oleh pengguna, dan hasil keluaran yang sesuai atau cocok dengan karakter *Unicode* pada *text area*.
- 4) “*Auto-learner*” adalah modul yang merupakan bagian dari *Recognizer* yang telah dipelajari kesalahan pengenalan yang telah dikoreksi oleh perantara *user*.

Langkah-langkah pengimplementasian API Gesture Android adalah sebagai berikut:

1) Membuat *Gesture Library* (*Template Library*).

Aplikasi Gesture Builder digunakan untuk membuat sebuah *Gesture library*. Pengembang dapat membuat pengenalan gerakan yang lebih sulit dan berbeda dengan pengenalan sederhana seperti *swipe* pada aplikasinya sendiri menggunakan Gesture Builder. Seperti contoh gesture pada **Gambar 2.8**.



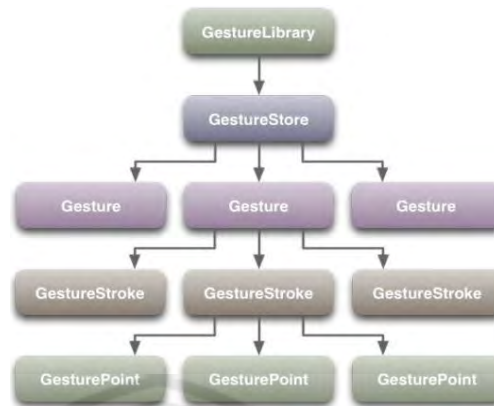
Gambar 2.8 Tampilan dari Gesture Builder (Casasola, 2012 : 14)

Gesture yang telah di buat akan memiliki identitas yaitu sebuah nama, nama tersebut yang akan membedakan antara *gesture* satu dengan *gesture* lainnya. Setiap kali melakukan penambahan atau pengeditan sebuah *gesture* di dalam Gesture Builder, file atau gesture tersebut akan tersimpan pada SD card dari emulator, /sdcard/gestures. File yang telah tersimpan akan berisi tentang deskripsi dari seluruh gesture yang telah disimpan atau dibuat, yang apabila suatu saat akan dibutuhkan untuk diletakan pada *package* yang telah disediakan dalam aplikasi yang akan dibuat dengan file directori, /res/raw (Casasola, 2012:14)

2) Memuat *Gesture Library*

Gestures (goresan atau tulisan tangan pada layar touch screen) yang sebelumnya telah dipresentasikan ulang pada *Android library* akan diberi nama yaitu *Gesture object*. *Gesture object* dapat terdiri dari sebuah *stroke* atau beberapa *stroke* yang dikenal dengan nama *GestureStroke*. *GestureStroke* terdiri dari beberapa urutan titik yang dikenal dengan nama *GesturePoints*. *Gesture* akan disimpan pada *GestureLibrary*, semua gesture akan disimpan dan

dipelihara oleh interface yang bernama GestureStore (Duta *et al*, 2012:684). Struktur dari GestureLibrary dapat dilihat pada **Gambar 2.9**.



Gambar 2.9 Struktur Gesture Library (Casasola, 2012 : 15)

Apabila *gesture* telah dibuat dan disimpan pada *GestureLibrary*, maka pengembang harus me-loadnya ke dalam aplikasi. Setelah aplikasi yang dibuat dapat menyimpan *library*, *library* akan dimuat oleh folder raw pada *package* aplikasi yang akan dibuat. *Library* yang ada di dalam aplikasi tidak akan dapat dimodifikasi, *library* tersebut hanya dapat dibaca.

3) Pengenalan *gestures*

Deskripsi singkat GUI yang dibutuhkan untuk proses pengenalan *gesture* terdiri dari : 1) *GestureOverlayView* : tempat pembuatan *gesture*, 2) *Canvas* : area dibawah tampilan *overlay*, 3) *Text Area*, 4) *Prediction Scores* dan *choices*.

Penambahan *GestureOverlayView* pada XML layout aplikasi yang akan dibuat dilakukan untuk melakukan pengenalan *gesture*. *GestureOverlayView* memiliki peran untuk berfungsi sebagai papan gambar sederhana untuk menggambar *gesture*. *GestureLibrary* akan memberikan bantuan berupa *list prediction* dengan memberikan sebuah nama yang telah diberikan pada Gesture Builder dan sebuah skor untuk mengenali hasil dari *gesture* yang telah dibuat.

2.5.3 API Voice Recognition

API Voice Recognition merupakan sebuah framework yang dimiliki oleh Google untuk mengenali suara dan merubahnya menjadi string (text). Framework ini memiliki kosakata yang terbatas kata-kata dan frase dalam pengenalan suara,

dan hanya dapat teridentifikasi jika *user* atau pengguna berbicara dengan sangat jelas.

Tipology *speech recognition* adalah sebagai berikut (Saini *et al*, 2013 : 132):

- 1) *Speaker Dependent* : Sebuah sistem memerlukan seorang pengguna untuk melatih sistem agar sesuai dengan keinginan penggunanya.
- 2) *Speaker Independent* : Sebuah sistem yang tidak memerlukan seorang pengguna untuk melatih sistem yang dimiliki, yaitu sistem yang dikembangkan untuk beroperasi kepada setiap pengguna.
- 3) *Isolated word recognizer* : Sebuah sistem yang dapat menerima satu kata dalam satu waktu. Sistem ini membuat pengguna agar berbicara secara alami secara terus menerus.
- 4) *Connected word system* : Sebuah sistem yang memungkinkan pengguna agar berbicara secara perlahan dan jelas pada setiap kata dengan jeda waktu yang singkat.
- 5) *Spontaneous recognition systems*: Sebuah sistem yang dapat membuat pengguna untuk berbicara secara spontan.

2.6 Rancang Bangun Aplikasi/ Perangkat Lunak

2.6.1 UML

Unified Modeling Language (UML) merupakan sebuah bahasa standart yang digunakan untuk menulis *blueprint* dari sebuah perangkat lunak. Aktivitas untuk menggambarkan, menetapkan, membangun dan mendokumentasikan artifak dari sebuah sistem perangkat lunak intensif dilakukan dengan menggunakan UML (Pressman, 2010 : 841). Beberapa diagram yang digunakan untuk membangun permodelan perangkat lunak adalah sebagai berikut :

1) *Use case diagram*

Use case diagram digunakan dengan tujuan untuk menentukan fungsi dan fitur dari sebuah perangkat lunak yang berasal dari prepektif pengguna. Interaksi antara seorang pengguna dengan sebuah sistem dengan cara mendefinisikan langkah-langkah yang diperlukan untuk mencapai sebuah tujuan tertentu akan dideskripsikan oleh sebuah *use case*. Segala keseluruhan permasalahan yang

dimiliki oleh pengguna dan hubungan yang dimiliki mereka akan digambarkan oleh sebuah *use case diagram*.

2) *Sequence diagram*

Komunikasi dinamis sebuah objek selama menjalankan tugas yang dimiliki akan ditunjukkan oleh *sequence diagram*. Pemanggilan metode yang menggunakan panah horizontal yang berasal dari *the caller* menuju kepada *the called*, yang telah diberikan label nama sesuai dengan nama metode dan opsional yang terdiri dari parameter, tipe parameter, dan tipe *return* akan ditunjukkan oleh *sequence diagram*. (Pressman, 2010 : 848-849).

3) *Activity diagram*

Perilaku dinamis sebuah sistem atau bagian dari sebuah sistem yang melalui aliran kontrol antara tindakan sistem digambarkan oleh *activity diagram*. *Activity diagram* hampir mirip dengan *flowchart* tetapi pada *activity diagram* dapat menunjukkan arus secara bersamaan. (Pressman, 2010 : 853).

2.6.2 Pengujian Perangkat Lunak

Verification dan *validation testing* (V&V) merupakan sebuah topik luas yang merupakan suatu elemen dari sebuah pengujian perangkat lunak. Verifikasi akan mengarah kepada serangkaian kegiatan yang dilakukan untuk memastikan sebuah perangkat lunak dapat melakukan suatu fungsi tertentu yang telah ditentukan. Sedangkan verifikasi akan mengarah kepada serangkaian kegiatan berbeda untuk memastikan sebuah perangkat lunak telah sesuai dengan kebutuhan dari sang pengguna (Pressman, 2010 : 450-451). Teknik pengujian perangkat lunak yang digunakan pada penelitian ini adalah *black-box testing* dan *user acceptance test*.

2.6.3 Black-box Testing

Sebuah metode pengujian yang fokus kepada kebutuhan fungsionalitas dari suatu perangkat lunak merupakan definisi dari *black-box testing*. Pada pengujian ini dilakukan upaya untuk menemukan kesalahan pada kategori : 1) adanya fungsi yang salah atau hilang, 2) terjadi error pada antarmuka, 3) terjadinya kesalahan pada struktur data atau pada proses akses database eksternal, 4) terjadinya error pada kinerja atau perilaku perangkat lunak, 5) terjadi kesalahan inisialisasi dan terminasi (Pressman, 2010 : 495).

2.6.4 User Acceptance Test

User Acceptance Test (UAT) adalah sebuah proses pengujian untuk melakukan verifikasi apakah fitur yang dibuat pada sebuah perangkat lunak sudah sesuai untuk pengguna atau *user* (perry, 2006). Pengujian ini berbeda dengan pengujian sistem (pengujian yang dilakukan untuk memastikan apakah tidak terjadi crash pada software). *User Acceptance Test* (UAT) pada umumnya dilakukan oleh *end-user*. Pada proses pengujian UAT penguji dan pengembang akan melakukan identifikasi dan memperbaiki masalah yang terjadi selama masih dalam tahap awal pengujian fungsionalitas pada tahap testing. Jenis UAT adalah sebagai berikut :

- 1) Alpha & Beta Testing.
- 2) Contract Acceptance Testing.
- 3) Regulation Acceptance Testing.
- 4) Operational Acceptance Testing.
- 5) Black-Box Testing.

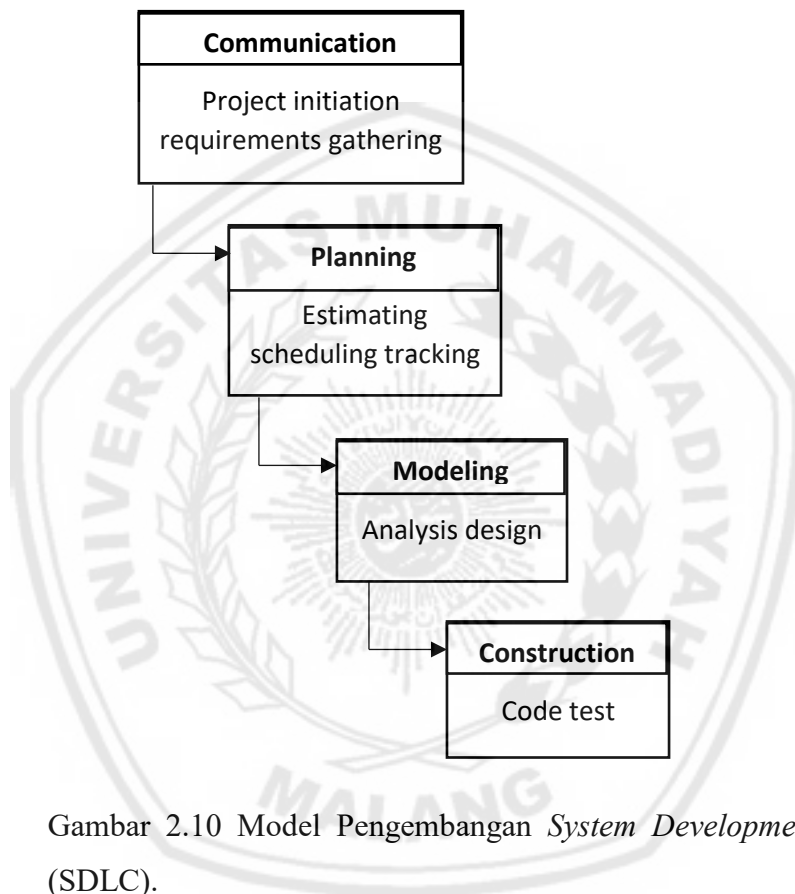
2.7 Metode Pengembangan Perangkat Lunak *System Development Life Cycle* (SDLC)

System Development Life Cycle (SDLC) atau dikenal juga dengan nama metode *waterfall* merupakan proses pembuatan sebuah sistem beserta model dan metodologi yang digunakan guna melakukan proses pengembangan kepada sistem-sistem tersebut (Pressman, 2010 : 39). Tahapan metode SDLC:

- 1) *Communication* : kegiatan awal yang dilakukan untuk mengumpulkan data atau informasi tentang kebutuhan yang diminta oleh pengguna, kegiatan ini dilakukan dengan cara melakukan komunikasi kepada pengguna.
- 2) *Planning* : kegiatan yang dilakukan untuk menetapkan rencana pengerjaan software meliputi mempelajari tentang sebuah konsep sistem dan permasalahan yang ingin diselesaikan, dan jadwal pengerjaan.
- 3) *Modeling* : kegiatan untuk membuat rancangan sebuah sistem teknologi guna menyelesaikan masalah yang sama. Pada proses ini dilakukan penerjemahan pada rancangan perangkat lunak sebelum dilakukan proses *coding*.
- 4) *Construction* : dilakukan pengkodean (coding), yaitu proses menerjemahkan desain yang telah dibuat kedalam bahasa yang dapat dikenali oleh komputer.

Setelah proses pengkodean selesai selanjutnya akan dilakukan proses yang bertujuan untuk menemukan kesalahan terhadap sistem yang telah dibuat, proses ini kita sering dikenal dengan proses *testing*.

Tahapan metode SDLC (*System Life Development Life Cycle*) memiliki 4 tahapan, urutan tahap metode SDLC dapat dilihat pada **Gambar 3.0**.



Gambar 2.10 Model Pengembangan *System Development Life Cycle* (SDLC).

2.8 Perangkat Lunak yang Digunakan

Dalam proses pembuatan atau proses implementasi aplikasi media pembelajaran huruf Hiragana dan Katakana menggunakan API Gesture dan API Voice Recognition peneliti menggunakan beberapa perangkat lunak, diantaranya adalah sebagai berikut :

2.8.1 Android Studio

Android Studio merupakan sebuah IDE untuk para Android Development yang diperkenalkan oleh Google pada sebuah acara Google I/O pada tahun 2013. Android Studio adalah hasil perkembangan dari Eclipse IDE. Android Studio adalah *software* IDE yang resmi untuk pengembangan aplikasi pada Android (Zapata, 2013).

Sebagai penerus yang merupakan hasil pengembangan pada Eclipse, maka Android Studio memiliki fitur-fitur baru. Perbedaan antara Android Studio dengan Eclipse adalah jika Eclipse menggunakan Ant sebagai build environment maka Android Studio menggunakan Gradle sebagai build environment, fitur-fitur baru yang dimiliki oleh Android Studio adalah sebagai berikut :

- 1) Dapat mem-build multiple APK.
- 2) Layout editor yang lebih bagus.
- 3) Built-in support untuk Google Cloud Platform, sehingga lebih mudah untuk melakukan integrasi dengan Google Cloud Messaging dan APP Engine.
- 4) Memiliki template yang support untuk Google Services dan berbagai macam tipe perangkat.
- 5) Menggunakan Gradle-based build system yang fleksibel.
- 6) Dapat melakukan import library langsung dari Maven Repository.

2.8.2 Android SDK (*Software Development Kit*)

Android SDK (*Software Development Kit*) merupakan bagian dari API (*Application Programming Interface*) berfungsi untuk menyediakan segala *tools* yang dibutuhkan untuk membangun aplikasi Android yang ditulis dengan menggunakan bahasa pemrograman Java. Pada Android SDK terdapat sebuah *compiler*, *debugger* dan sebuah *device emulator* yang digunakan sebagai mesin virtual untuk menjalankan aplikasi Android (Zapata, 2013).

2.8.3 ADT (*Android Development Tools*)

ADT (*Android Development Tools*) merupakan sebuah *plug-in* untuk Eclipse IDE yang dirancang untuk membuat sebuah aplikasi Android. ADT membuat Eclipse dapat menciptakan proyek Android baru, membuat aplikasi UI, menambahkan berbagai komponen yang berdasarkan pada *framework* Android API, melakukan *debugs* aplikasi menggunakan Android SDK *tools*, dan dan mengexport aplikasi menjadi file .apk, sehingga aplikasi dapat diinstal pada *smartphone*. Melakukan pembuatan aplikasi Android pada Eclipse dengan ADT sangat di rekomendasikan karena dengan menggunakan ADT pada Eclipse merupakan langkah awal tercepat dalam proses pembuatan aplikasi Android (Zapata 2013).

